Lab Exercise: Lambda Expressions

INTERTECH

Instructors Who Consult. | **Consultants** Who Teach.

About Intertech

Thank you for choosing Intertech for your training from Udemy. The next page of this document will get you started with your lab if you'd like to skip ahead. Below is a brief overview of Intertech as well as a promo code for you for future live Intertech trainings.

Intertech (www.intertech.com) is the largest combined software developer **training** and **consulting** firm in Minnesota. Our unique blend of training, consulting, and mentoring has empowered technology teams in small businesses, Fortune 500 corporations and government agencies since 1991.

Our training organization offers live in-classroom and online deliveries, private on-site deliveries, and on-demand options such as the course in which you've enrolled from Udemy. We cover a broad spectrum of .NET, Java, Agile/Scrum, Web Development, and Mobile technologies. See more information on our training and search for courses by **clicking here**.

As a Udemy customer, you can save on any live in-classroom, live online, or onsite training with Intertech as well. Just use promo code "Udemy_Labs" when enrolling **and** save 35% on the course price.

We appreciate you choosing Intertech on Udemy for this course!

Lab Exercise

Lambda Expressions

Lambda expressions give us the ability to quickly define an implementation for a functional interface. This feature adds flexibility to an app, with the least amount of code possible!

In this lab, you will explore an array to store MyDate objects.

Specifically, in this lab you will:

- Create a functional interface
- Implement a lambda expression

Scenario

Certain orders can be flagged as “priority orders,” where they will be placed ahead in the shipping queue and sent via next day shipping. The criteria that determines what orders should be prioritized may change at any point. However, when the criteria chances, it changes for all orders (rather than allowing each order to determine its own rules). To handle this use case, a functional interface called Rushable defines a single “isRushable” method. The method is passed an orderDate and orderAmount. It is up to the implementation to decide if they will use either, both, or none of these parameters.

Step 1: Create the Rushable functional interface.

1.1 In the com.acme.domain package, create an interface called Rushable.

1.2 Add the following method to it:

```
public abstract boolean isRushable(MyDate orderDate, double
amount);
```

Step 2: Add a private static Rushable variable to Order.java with a getter/setter pair.

```
private static Rushable rushable;
public static Rushable getRushable()
{
  return rushable;
}
public static void setRushable(Rushable rushable)
{
  Order.rushable = rushable;
}
```

Step 3: Create the isPriorityOrder method which delegates to the Rushable object.

```
public boolean isPriorityOrder()
{
   boolean priorityOrder = false;
   if( rushable != null ) {
         priorityOrder = rushable.isRushable(orderDate, 
orderAmount);
   }
   return priorityOrder;
}
```

***Step 4:	On Your Own: In TestOrders.java pass in a Lambda expression to setRushable which returns true if the orderAmount is over 1500.***

***Step 5:	In TestOrders.java, call isPriorityOrder() on the anvil and balloon and make sure they return "true," and "false" respectively.***

```
System.out.println("Anvil isPriorityOrder: " + 
anvil.isPriorityOrder()); // true
System.out.println("Balloons isPriorityOrder: " + 
balloons.isPriorityOrder()); // false
```

Lab Solutions

Rushable.java

```
package com.acme.domain;

import com.acme.utils.MyDate;

public interface Rushable
{
  public abstract boolean isRushable(MyDate orderDate, double
amount);
}
```

TestOrders.java

```
package com.acme.testing;

import com.acme.domain.Good.UnitOfMeasureType;
import com.acme.domain.Order;
import com.acme.domain.Service;
import com.acme.domain.Solid;
import com.acme.utils.MyDate;

public class TestOrders
{
  public static void main( String[] args )
  {
    MyDate date1 = new MyDate( 1, 20, 2008 );
    Solid s1 = new Solid( "Acme Anvil", 1668, 0.3,
        UnitOfMeasureType.CUBIC_METER, false, 500, 0.25, 0.3 );
    Order anvil = new Order( date1, 2000.00, "Wile E Coyote",
s1, 10 );
    MyDate date2 = new MyDate( 4, 10, 2008 );
    Solid s2 = new Solid( "Acme Balloon", 1401, 15,
        UnitOfMeasureType.CUBIC_FEET, false, 10, 5, 5 );
    Order balloons = new Order( date2, 1000.00, "Bugs Bunny",
s2, 125 );

  Order.setRushable((orderDate, orderAmount) -> orderAmount >
1500 );
  System.out.println("Anvil isPriorityOrder: " +
anvil.isPriorityOrder());
  System.out.println("Balloons isPriorityOrder: " +
balloons.isPriorityOrder());

        ....
    }
}
```

Order.java

```
package com.acme.domain;

import com.acme.utils.MyDate;

public class Order {
   private MyDate orderDate;
   private double orderAmount = 0.00;
   private String customer;
   private Product product;
   private int quantity;
   private static Rushable rushable;

   public static Rushable getRushable() {
         return rushable;
   }

   public static void setRushable(Rushable rushable) {
         Order.rushable = rushable;
   }

   public boolean isPriorityOrder()
   {
         boolean priorityOrder = false;
         if( rushable != null ) {
               priorityOrder = rushable.isRushable(orderDate,
orderAmount);
         }
         return priorityOrder;
   }

     ....
}
```

Give Intertech a call at **1.800.866.9884** or visit Intertech's website.